

Отчёт по лабораторной работе №12

Дисциплина: Операционные системы

Мишина Анастасия Алексеевна

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	8
4	Выполнение лабораторной работы	9
5	Выполнение заданий самостоятельной работы	14
6	Вывод	16

Список иллюстраций

4.1	Скрипт к заданию 1.	9
4.2	Результат выполнения скрипта 1.	10
4.3	Просмотр каталога /usr/share/man/man1.	10
4.4	Скрипт к заданию 2.	11
4.5	Выполнения скрипта 2.	11
4.6	Результат выполнения скрипта 2.	12
4.7	Скрипт к заданию 3.	12
4.8	Результат выполнения скрипта 3.	13

Список таблиц

1 Цель работы

Изучить основы программирования в оболочке ОС UNIX. Научиться писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

2 Задание

1. Написать командный файл, реализующий упрощённый механизм семафоров. Командный файл должен в течение некоторого времени t_1 дожидаться освобождения ресурса, выдавая об этом сообщение, а дождавшись его освобождения, использовать его в течение некоторого времени $t_2 < t_1$, также выдавая информацию о том, что ресурс используется соответствующим командным файлом (процессом). Запустить командный файл в одном виртуальном терминале в фоновом режиме, перенаправив его вывод в другой, в котором также запущен этот файл, но не фоновом, а в привилегированном режиме. Доработать программу так, чтобы имелась возможность взаимодействия трёх и более процессов.
2. Реализовать команду `man` с помощью командного файла. Изучите содержимое каталога `/usr/share/man/man1`. В нем находятся архивы текстовых файлов, содержащих справку по большинству установленных в системе программ и команд. Каждый архив можно открыть командой `less` сразу же просмотрев содержимое справки. Командный файл должен получать в виде аргумента командной строки название команды и в виде результата выдавать справку об этой команде или сообщение об отсутствии справки, если соответствующего файла нет в каталоге `man1`.

3. Используя встроенную переменную \$RANDOM, напишите командный файл, генерирующий случайную последовательность букв латинского алфавита. Учтите, что \$RANDOM выдаёт псевдослучайные числа в диапазоне от 0 до 32767.

3 Теоретическое введение

Преимущества и недостатки Bash:

Многие языки программирования намного удобнее и понятнее для пользователя. Например, Python более быстр, так как компилируется байтами. Однако главное преимущество Bash – его повсеместное распространение. Более того, Bash позволяет очень легко работать с файловой системой без лишних конструкций (в отличие от других языков программирования). Но относительно таких `bash` очень сжат. То есть, например, C имеет гораздо более широкие возможности для разработчика.

4 Выполнение лабораторной работы

1. Я написала командный файл, реализующий упрощенный механизм семафоров (рис. [4.1]).

```
lockfile="./locking.file"

exec {fn}>"$lockfile"
if test -f "$lockfile"
then
    while [ 1 != 0 ]
    do
        if flock -n ${fn}
        then
            echo "file was locked"
            sleep 4
            echo "unlocking"
            flock -u ${fn}
        else
            echo "file was unlocked"
            sleep 3
        fi
    done
fi
```

Рис. 4.1: Скрипт к заданию 1.

Затем я добавила право на исполнение файла и выполнила его (рис. [4.2]).

```
[aamishina@fedora lab12]$ vi lab12_1
[aamishina@fedora lab12]$ chmod +x lab12_1
[aamishina@fedora lab12]$ ./lab12_1
file was locked
unlocking
```

Рис. 4.2: Результат выполнения скрипта 1.

2. Я просмотрела содержимое каталога /usr/share/man/man1 (рис. [4.3]).

```
[aamishina@fedora lab12]$ ls /usr/share/man/man1
:~.1.gz
'~.1.gz'
a2ping.1.gz
ab.1.gz
abrt.1.gz
abrt-action-analyze-backtrace.1.gz
abrt-action-analyze-c.1.gz
abrt-action-analyze-ccpp-local.1.gz
abrt-action-analyze-core.1.gz
abrt-action-analyze-java.1.gz
abrt-action-analyze-oops.1.gz
abrt-action-analyze-python.1.gz
abrt-action-analyze-vmcore.1.gz
abrt-action-analyze-vulnerability.1.gz
```

Рис. 4.3: Просмотр каталога /usr/share/man/man1.

Я написала командный файл, позволяющий реализовать команду man с помощью команды less, которая выдает содержимое справки по команде (рис. [4.4]).

```
command=""

while getopts :m: opt
do
    case $opt in
        m) command="$OPTARG";;
        esac
    done

    if test -f "/usr/share/man/man1/$command.1.gz"
    then less /usr/share/man/man1/$command.1.gz
    else
    echo "No such command!"
    fi
fi
~
```

Рис. 4.4: Скрипт к заданию 2.

Затем я добавила право на исполнение файла и выполнила его (рис. [4.5]), (рис. [4.6]).

```
[aamishina@fedora lab12]$ vi lab12_2
[aamishina@fedora lab12]$ chmod +x lab12_2
[aamishina@fedora lab12]$ ./lab12_2 -m pwd
[aamishina@fedora lab12]$
```

Рис. 4.5: Выполнения скрипта 2.

```
PWD(1)                                User Commands                                PWD(1)

ESC[1mNAMEESC[0m
    pwd - print name of current/working directory

ESC[1mSYNOPSISESC[0m
    ESC[1mpwd ESC[22m[ESC[4mOPTIONESC[24m]...

ESC[1mDESCRIPTIONESC[0m
    Print the full filename of the current working directory.

    ESC[1m-LESC[22m, ESC[1m--logicalESC[0m
        use PWD from environment, even if it contains symlinks

    ESC[1m-BESC[22m, ESC[1m--physicalESC[0m
```

Рис. 4.6: Результат выполнения скрипта 2.

3. Я написала командный файл, который генерировал случайную последовательность букв латинского алфавита (рис. [4.7]).

```
echo "password:"
cat /dev/urandom | tr -dc "a-zA-Z0-9" | fold -w 15 | head -n 1
```

Рис. 4.7: Скрипт к заданию 3.

Затем я добавила право на исполнение файла и выполнила его (рис. [4.8]).

```
[aamishina@fedora lab12]$ vi lab12_3
[aamishina@fedora lab12]$ chmod +x lab12_3
[aamishina@fedora lab12]$ ./lab12_3
password:
8wWNVPxZxHj4l57
[aamishina@fedora lab12]$ ./lab12_3
password:
N7Don8QN76sGpJc
[aamishina@fedora lab12]$ ./lab12_3
password:
y7MIgJPGsLbLhaq
[aamishina@fedora lab12]$
```

Рис. 4.8: Результат выполнения скрипта 3.

5 Выполнение заданий самостоятельной работы

Контрольные вопросы

1. Найдите синтаксическую ошибку в следующей строке: `1 while [$1 != "exit"]`

Между скобками должны быть пробелы, иначе символы в скобках и сами скобки будут восприняты как один элемент.

2. Как объединить (конкатенация) несколько строк в одну?

```
cat file.txt | xargs
```

3. Найдите информацию об утилите `seq`. Какими иными способами можно реализовать её функционал при программировании на `bash`?

Команда `seq` выводит последовательность целых или действительных чисел, подходящую для передачи в другие программы. Реализовать ее функционал можно командой `for n in {1..5} do done`

4. Какой результат даст вычисление выражения $\$(10/3)$? Вычисление этого выражения даст результат 3
5. Укажите кратко основные отличия командной оболочки `zsh` от `bash`.

Zsh очень сильно упрощает работу. Но существуют различия. Например, в zsh после for обязательно вставлять пробел, нумерация массивов в zsh начинается с 1. Если вы собираетесь писать скрипт, который будет запускать множество разработчиков, то рекомендуется Bash. Если скрипты вам не нужны - Zsh.

6. Проверьте, верен ли синтаксис данной конструкции `1 for ((a=1; a <= LIMIT; a++))`

Да, этот синтаксис верен.

7. Сравните язык `bash` с какими-либо языками программирования. Какие преимущества у `bash` по сравнению с ними? Какие недостатки?

Многие языки программирования намного удобнее и понятнее для пользователя. Например, Python более быстр, так как компилируется байтами. Однако главное преимущество Bash – его повсеместное распространение. Более того, Bash позволяет очень легко работать с файловой системой без лишних конструкций (в отличие от других языков программирования). Но относительно таких `bash` очень сжат. То есть, например, C имеет гораздо более широкие возможности для разработчика.

6 Вывод

В ходе выполнения лабораторной работы я изучила основы программирования в оболочке ОС UNIX, а также научилась писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.